Phrase Based Decoding using a Discriminative Model

Prasanth Kolachina
LTRC, IIIT-Hyderabad
{prasanth_k}@research.iiit.ac.in

Sriram Venkatapathy
LTRC, IIIT-Hyderabad
{sriram}@research.iiit.ac.in

Srinivas Bangalore
AT&T Labs-Research, NY
{srini}@research.att.com

Sudheer Kolachina
LTRC, IIIT-Hyderabad
{sudheer.kpg08}@research.iiit.ac.in

Avinesh PVS
LTRC, IIIT-Hyderabad
{avinesh}@research.iiit.ac.in

Abstract

In this paper, we present an approach to statistical machine translation that combines the power of a discriminative model (for training a model for Machine Translation), and the standard beam-search based decoding technique (for the translation of an input sentence). A discriminative approach for learning lexical selection and reordering utilizes a large set of feature functions (thereby providing the power to incorporate greater contextual and linguistic information), which leads to an effective training of these models. This model is then used by the standard state-of-art Moses decoder (Koehn et al., 2007) for the translation of an input sentence.

We conducted our experiments on Spanish-English language pair. We used maximum entropy model in our experiments. We show that the performance of our approach (using simple lexical features) is comparable to that of the state-of-art statistical MT system (Koehn et al., 2007). When additional syntactic features (POS tags in this paper) are used, there is a boost in the performance which is likely to improve when richer syntactic features are incorporated in the model.

1 Introduction

The popular approaches to machine translation use the generative IBM models for training (Brown et al., 1993; Och et al., 1999). The parameters for these models are learnt using the standard EM Algorithm. The parameters used in these models are extremely restrictive, that is, a simple, small and closed set of feature functions is used to represent the translation process. Also, these feature functions are local and are word based. In spite of these limitations, these models perform very well for the task of word-alignment because of the restricted search space. However, they perform poorly during decoding (or translation) because of their limitations in the context of a much larger search space.

To handle the contextual information, phrase-based models were introduced (Koehn et al., 2003). The phrase-based models use the word alignment information from the IBM models and train source-target phrase pairs for lexical selection (phrase-table) and distortions of source phrases (reordering-table). These models are still relatively local, as the target phrases are tightly associated with their corresponding source phrases. In contrast to a phrase-based model, a discriminative model has the power to integrate much richer contextual information into the training model. Contextual information is extremely useful in making lexical selections of higher quality, as illustrated by the models for Global Lexical Selection (Bangalore et al., 2007; Venkatapathy and

Proceedings of SSST-4, Fourth Workshop on Syntax and Structure in Statistical Translation, pages 34–42,
COLING 2010, Beijing, August 2010.

Bangalore, 2009).

However, the limitation of global lexical selection models has been sentence construction. In global lexical selection models, lattice construction and scoring (LCS) is used for the purpose of sentence construction (Bangalore et al., 2007; Venkatapathy and Bangalore, 2009). In our work, we address this limitation of global lexical selection models by using an existing state-of-art decoder (Koehn et al., 2007) for the purpose of sentence construction. The translation model used by this decoder is derived from a discriminative model, instead of the usual phrase-table and reordering-table construction algorithms. This allows us to use the effectiveness of an existing phrase-based decoder while retaining the advantages of the discriminative model. In this paper, we compare the sentence construction accuracies of lattice construction and scoring approach (see section 4.1 for LCS Decoding) and the phrase-based decoding approach (see section 4.2).

Another advantage of using a discriminative approach to construct the phrase table and the reordering table is the flexibility it provides to incorporate linguistic knowledge in the form of additional feature functions. In the past, factored phrase-based approaches for Machine Translation have allowed the use of linguistic feature functions. But, they are still bound by the locality of context, and definition of a fixed structure of dependencies between the factors (Koehn and Hoang, 2007). Furthermore, factored phrase-based approaches place constraints both on the type and number of factors that can be incorporated into the training. In this paper, though we do not extensively test this aspect, we show that using syntactic feature functions does improve the performance of our approach, which is likely to improve when much richer syntactic feature functions (such as information about the parse structure) are incorporated in the model.

As the training model in a standard phrase-based system is relatively impoverished with respect to contextual/linguistic information, integration of the discriminative model in the form of phrase-table and reordering-table with the phrase-based decoder is highly desirable. We propose to do this by defining sentence specific tables. For example, given a source sentence s, the phrase-table contains all the possible phrase-pairs conditioned on the context of the source sentence s.

In this paper, the key contributions are,

1. We combine a discriminative training model with a phrase-based decoder. We obtained comparable results with the state-of-art phrase-based decoder.

2. We evaluate the performance of the lattice construction and scoring (LCS) approach to decoding. We observed that even though the lexical accuracy obtained using LCS is high, the performance in terms of sentence construction is low when compared to phrase-based decoder.

3. We show that the incorporation of syntactic information (POS tags) in our discriminative model boosts the performance of translation. In future, we plan to use richer syntactic feature functions (which the discriminative approach allows us to incorporate) to evaluate the approach.

The paper is organized in the following sections. Section 2 presents the related work. In section 3, we describe the training of our model. In section 4, we present the decoding approaches (both LCS and phrase-based decoder). We describe the data used in our experiments in section 5. Section 6 consists of the experiments and results. Finally we conclude the paper in section 7.

2 Related Work

In this section, we present approaches that are directly related to our approach. In Direct Translation Model (DTM) proposed for statistical machine translation by (Papineni et al., 1998; Och and Ney, 2002), the authors present a discriminative set-up for natural language understanding (and MT). They use a slightly modified equation (in comparison to IBM models) as shown in equation 1. In equation 1, they consider the translation model from $f \rightarrow e$ ($p(e|f)$), instead of the theoretically sound (after the application of Bayes' rule), $e \rightarrow f$ ($p(f|e)$) and use grammatical features such as the presence of equal number of verbs forms etc.

$$\hat{e} = \arg\max_{e} p_{TM}(e|f) * p_{LM}(e) \quad (1)$$

In their model, they use generic feature functions such as language model, cooccurence features such as presence of a lexical relationship in the lexicon. Their search algorithm limited the use of complex features.

Direct Translation Model 2 (DTM2) (Ittycheriah and Roukos, 2007) expresses the phrase-based translation task in a unified log-linear probabilistic framework consisting of three components:

1. a prior conditional distribution P_0
2. a number of feature functions $\Phi_i()$ that capture the effects of translation and language model
3. the weights of the features λ_i that are estimated using MaxEnt training (Berger et al., 1996) as shown in equation 2.

$$Pr(e|f) = \frac{P_0(e,j|f)}{Z} exp \sum_i \lambda_i \Phi_i(e,j,f) \quad (2)$$

In the above equation, j is the skip reordering factor for the phrase pair captured by $\Phi_i()$ and represents the jump from the previous source word. Z represents the per source sentence normalization term (Hassan et al., 2009). While a uniform prior on the set of futures results in a *maximum entropy* model, choosing other priors output a *minimum divergence* models. Normalized phrase count has been used as the prior P_0 in the DTM2 model.
The following decision rule is used to obtain optimal translation.

$$\begin{aligned} \hat{e} &= \arg\max_{e} Pr(e|f) \\ &= \arg\max_{e} \sum_{m=1}^{M} \lambda_m \Phi_m(f,e) \end{aligned} \quad (3)$$

The DTM2 model differs from other phrase-based SMT models in that it avoids the redundancy present in other systems by extracting from a word aligned parallel corpora a set of minimal phrases such that no two phrases overlap with each other (Hassan et al., 2009).

The decoding strategy in DTM2 (Ittycheriah and Roukos, 2007) is similar to a phrase-based decoder except that the score of a particular translation block is obtained from the maximum entropy model using the set of feature functions. In our approach, instead of providing the complete scoring function ourselves, we compute the parameters needed by a phrase based decoder, which in turn uses these parameters appropriately. In comparison with the DTM2, we also use minimal non-overlapping blocks as the entries in the phrase table that we generate.

Xiong et al. (2006) present a phrase reordering model under the ITG constraint using a maximum entropy framework. They model the reordering problem as a two-class classification problem, the classes being *straight* and *inverted.* The model is used to merge the phrases obtained from translating the segments in a source sentence. The decoder used is a hierarchical decoder motivated from the CYK parsing algorithm employing a beam search algorithm. The maximum entropy model is presented with features extracted from the blocks being merged and probabilities are estimated using the log-linear equation shown in (4). The work in addition to lexical features and collocational features, uses an additional metric called the information gain ratio (IGR) as a feature. The authors report an improvement of 4% BLEU score over the traditional distance based distortion model upon using the lexical features alone.

$$p_\lambda(y|x) = \frac{1}{Z_\lambda(x)} exp(\sum_i \lambda_i \Phi_i(x,y)) \quad (4)$$

3 Training

The training process of our approach has two steps:

1. training the discriminative models for translation and reordering.
2. integrating the models into a phrase based decoder.

The input to our training step are the word-alignments between source and target sentences obtained using GIZA++ (implementation of IBM, HMM models).

3.1 Training discriminative models

We train two models, one to model the translation of source blocks, and the other to model the reordering of source blocks. We call the translation model a 'context dependent block translation model' for two reasons.

1. It is concerned with the translation of minimal phrasal units called blocks.
2. The context of the source block is used during its translation.

The word alignments are used to obtain the set of possible target blocks, and are added to the target vocabulary. A target block b is a sequence of n words that are paired with a sequence of m source words (Ittycheriah and Roukos, 2007). In our approach, we restrict ourselves to target blocks that are associated with only one source word. However, this constraint can be easily relaxed.

Similarly, we call the reordering model, a 'context dependent block distortion model'. For training, we use the maximum entropy software library Llama presented in (Haffner, 2006).

3.1.1 Context Dependent Block Translation Model

In this model, the goal is to predict a target block given the source word and contextual and syntactic information. Given a source word and its lexical context, the model estimates the probabilities of the presence or absence of possible target blocks (see Figure 1).

The probabilities of the candidate target blocks are obtained from the maximum entropy model. The probability p_{e_i} of a candidate target block e_i is estimated as given in equation 5

$$p_{e_i} = P(true|e_i, f_j, C) \tag{5}$$

where f_j is the source word corresponding to e_i and C is its context.

Using the maximum entropy model, binary classifiers are trained for every target block in the vocabulary. These classifiers predict if a particular target block should be present given the source word and its context. This model is similar to the global lexical selection (GLS) model described in (Bangalore et al., 2007; Venkatapathy and Bangalore, 2009) except that in GLS, the predicted target blocks are not associated with any particular source word unlike the case here.

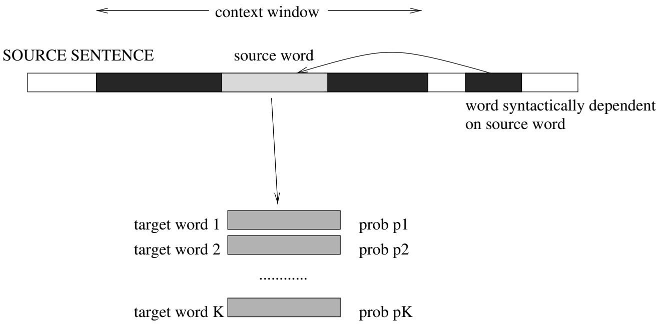

Figure 1: Word prediction model

For the set of experiments in this paper, we used a context of size 6, containing three words to the left and three words to the right. We also used the POS tags of words in the context window as features. In future, we plan to use the words syntactically dependent on a source word as global context(shown in Figure 1).

3.1.2 Context Dependent Block Distortion Model

An IBM model 3 like distortion model is trained to predict the relative position of a source word in the target given its context. Given a source word and its context, the model estimates the probability of particular relative position being an appropriate position of the source word in the target (see Figure 2).

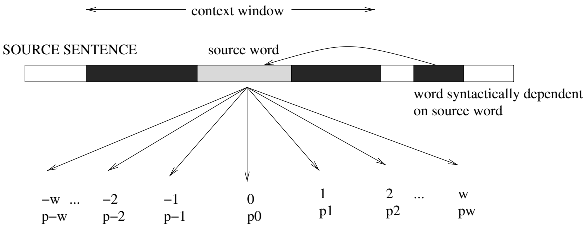

Figure 2: Position prediction model

Using a maximum entropy model similar to the one described in the context dependent block translation model, binary classifiers are trained for every possible relative position in the target. These classifiers output a probability distribution over various relative positions given a source word and its context.

The word alignments in the training corpus are used to train the distortion model. While computing the relative position, the difference in sentence lengths is also taken into account. Hence, the relative position of the target block located at position i corresponding to the source word located at position j is given in equation 6.

$$r = \textbf{round}(i * \frac{m}{n} - j) \quad (6)$$

where, m is the length of source sentence and n is the number of target blocks. **round** is the function to compute the nearest integer of the argument. If the source word is not aligned to any target word, a special symbol 'INF' is used to indicate such a case. In our model, this symbol is also a part of the target distribution.

The features used to train this model are the same as those used for the block translation model. In order to use further lexical information, we also incorporated information about the target word for predicting the distribution. The information about possible target words is obtained from the 'context dependent block translation model'. The probabilities in this case are measured as shown in equation 7

$$p_{r,e_i} = P(true|r, e_i, f_j, C) \quad (7)$$

3.2 Integration with phrase-based decoder

The discriminative models trained are sentence specific, i.e. the context of the sentence is used to make predictions in these models. Hence, the phrase-based decoder is required to use information specific to a source sentence. In order to handle this issue, a different phrase-table and reordering-table are constructed for every input sentence. The phrase-table and reordering-table are constructed using the discriminative models trained earlier.

In Moses (Koehn et al., 2007), the phrase-table contains the source phrase, the target phrase and the various scores associated with the phrase pair such as phrase translation probability, lexical weighting, inverse phrase translation probability, etc.[1]

In our approach, given a source sentence, the following steps are followed to construct the phrase table.

1. Extract source blocks ('words' in this work)

2. Use the 'context dependent block translation model' to predict the possible target blocks. The set of possible blocks can be predicted using two criteria, (1) Probability threshold, and (2) K-best. Here, we use a threshold value to prune the set of possible candidates in the target vocabulary.

3. Use the prediction probabilities to assign scores to the phrase pairs.

A similar set of steps is used to construct the reordering-table corresponding to an input sentence in the source language.

4 Decoding

4.1 Decoding with LCS Decoder

The lattice construction and scoring algorithm, as the name suggests, consists of two steps,

1. Lattice construction

   In this step, a lattice representing various possible target sequences is obtained. In the approach for global lexical selection (Bangalore et al., 2007; Venkatapathy and Bangalore, 2009), the input to this step is a bag of words. The bag of words is used to construct an initial sequence (a single path lattice). To this sequence, deletion arcs are added to incorporate additional paths (at a cost) that facilitate deletion of words in the initial sequence. This sequence is permuted using a permutation window in order to construct a lattice representing possible sequences. The permutation window is used to control the search space.

   In our experiments, we used a similar process for sentence construction. Using the context dependent block translation algorithm,

[1] http://www.statmt.org/moses/?n=FactoredTraining.ScorePhrases

we obtain a number of translation blocks for every source word. These blocks are interconnected in order to obtain the initial lattice (see figure 3).

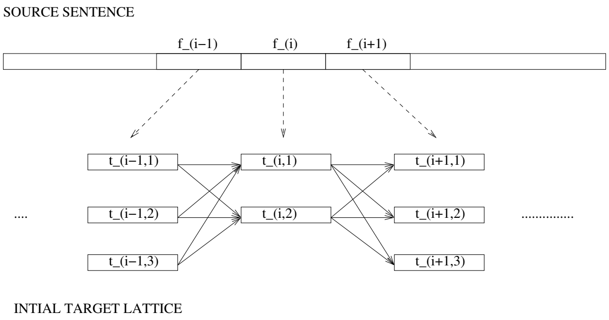

Figure 3: Lattice Construction

To control deletions at various source positions, deletion nodes may be added to the initial lattice. This lattice is permuted using a permutation window to construct a lattice representing possible sequences. Hence, the parameters that dictate lattice construction are, (1) Threshold for lexical selection, (2) Using deletion arcs or not, and (3) Permutation window.

2. Scoring

   In this step, each of the paths in the lattice constructed in the earlier step is scored using a language model (Haffner, 2006), which is same as the one used in the sentence construction in global lexical selection models. It is to be noted that we do not use the discriminative reordering model in this decoder, and only the language model is used to score various target sequences.

The path with the lowest score is considered the best possible target sentence for the given source sentence. Using this decoder, we conducted experiments on the development set by varying threshold values and the size of the permutation window. The best parameter values obtained using the development set were used for decoding the test corpus.

4.2 Decoding with Moses Decoder

In this approach, the phrase-table and the reordering-table are constructed using the discriminative model for every source sentence (see section 3.2). These tables are then used by the state-of-art Moses decoder to obtain corresponding translations.

The various training and decoding parameters of the discriminative model are computed by exhaustively exploring the parameter space, and correspondingly measuring the output quality on the development set. The best set of parameters were used for decoding the sentences in the test corpus. We modified the weights assigned by MOSES to the translation model, reordering model and language model. Experiments were conducted by performing pruning on the options in the phrase table and by using the word penalty feature in MOSES.

We trained a language model of order 5 built on the entire EUROPARL corpus using the SRILM package. The method uses improved Kneser-Ney smoothing algorithm (Chen and Goodman, 1999) to compute sequence probabilities.

5 Dataset

The experiments were conducted on the Spanish-English language pair. The latest version of the Europarl corpus(version-5) was used in this work. A small set of 200K sentences was selected from the training set to conduct the experiments. The test and development sets containing 2525 sentences and 2051 sentences respectively were used, without making any changes.

Corpus	No. of sentences	Source	Target
Training	200000	59591	36886
Testing	2525	10629	8905
Development	2051	8888	7750
Monolingual English (LM)	200000	n.a	36886

Table 1: Corpus statistics for Spanish-English corpus.

6 Experiments and Results

The output of our experiments was evaluated using two metrics, (1) BLEU (Papineni et al., 2002), and (2) Lexical Accuracy (LexAcc). Lexical accuracy measures the similarity between the unordered bag of words in the reference sentence against the unordered bag of words in the hypothesized translation. Lexical accuracy is a measure of the fidelity of lexical transfer from the source to the target sentence, independent of the syntax of the target language (Venkatapathy and Bangalore, 2009). We report lexical accuracies to show the performance of LCS decoding in comparison with the baseline system.

We first present the results of the state-of-art phrase-based model (Moses) trained on a parallel corpus. We treat this as our baseline. The reordering feature used is msd-bidirectional, which allows for all possible reorderings over a specified distortion limit. The baseline accuracies are shown in table 2.

Corpus	BLEU	Lexical Accuracy
Development	0.1734	0.448
Testing	0.1823	0.492

Table 2: Baseline Accuracy

We conduct two types of experiments to test our approach.

1. Experiments using lexical features (see section 6.1), and
2. Experiments using syntactic features (see section 6.2).

6.1 Experiments using Lexical Features

In this section, we present results of our experiments that use only lexical features. First, we measure the translation accuracy using LCS decoding. On the development set, we explored the set of decoding parameters (as described in section 4.1) to compute the optimal parameter values. The best lexical accuracy obtained on the development set is **0.4321** and the best BLEU score obtained is **0.0923** at a threshold of 0.17 and a permutation window size of value 3. The accuracies corresponding to a few other parameter values are shown in Table 3.

On the test data, we obtained a lexical accuracy of **0.4721** and a BLEU score of **0.1023**. As we can observe, the BLEU score obtained using the LCS decoding technique is low when compared to the BLEU score of the state-of-art system. However, the lexical accuracy is comparable to the lexical accuracy of Moses. This shows that the discriminative model provides good lexical selection, while the sentence construction technique does not perform as expected.

Threshold	Perm. Window	LexAcc	BLEU
0.16	3	0.4274	0.0914
0.17	3	0.4321	0.0923
0.18	3	0.4317	0.0918
0.16	4	0.4297	0.0912
0.17	4	0.4315	0.0915

Table 3: Lexical Accuracies of Lattice-Output using lexical features alone for various parameter values

Next, we present the results of the Moses based decoder that uses the discriminative model (see section 3.2). In our experiments, we did not use MERT training for tuning the Moses parameters. Rather, we explore a set of possible parameter values (i.e. weights of the translation model, reordering model and the language model) to check the performance. We show the BLEU scores obtained on the development set using Moses decoder in Table 4.

Reordering weight(d)	LM weight(l)	Translation weight(t)	BLEU
0	0.6	0.3	0.1347
0	0.6	0.6	0.1354
0.3	0.6	0.3	0.1441
0.3	0.6	0.6	0.1468

Table 4: BLEU for different weight values using lexical features only

On the test set, we obtained a BLEU score of **0.1771**. We observe that both the lexical accuracy and the BLEU scores obtained using the discriminative training model combined with the Moses decoder are comparable to the state-of-art results. The summary of the results obtained using three approaches and lexical feature functions is presented in Table 5.

6.2 Experiments using Syntactic Features

In this section, we present the effect of incorporating syntactic features using our model on the

Approach	BLEU	LexAcc
State-of-art(MOSES)	0.1823	0.492
LCS decoding	0.1023	0.4721
Moses decoder trained using a discriminative model	0.1771	0.4841

Table 5: Translation accuracies using lexical features for different approaches

translation accuracies. Table 6 presents the results of our approach that uses syntactic features at different parameter values. Here, we can observe that the translation accuracies (both LexAcc and BLEU) are better than the model that uses only lexical features.

Reordering weight(d)	**LM weight(l)**	**Translation weight(t)**	BLEU
0	0.6	0.3	0.1661
0	0.6	0.6	0.1724
0.3	0.6	0.3	0.1780
0.3	0.6	0.6	0.1847

Table 6: BLEU for different weight values using syntactic features

Table 7 shows the comparative performance of the model using syntactic as well as lexical features against the one with lexical features functions only.

Model	**BLEU**	**LexAcc**
Lexical features	0.1771	0.4841
Lexical+Syntactic features	**0.201**	**0.5431**

Table 7: Comparison between translation accuracies from models using syntactic and lexical features

On the test set, we obtained a BLEU score of **0.20** which is an improvement of **2.3** points over the model that uses lexical features alone. We also obtained an increase of **6.1%** in lexical accuracy using this model with syntactic features as compared to the model using lexical features only.

7 Conclusions and Future Work

In this paper, we presented an approach to statistical machine translation that combines the power of a discriminative model (for training a model for Machine Translation), and the standard beam-search based decoding technique (for the translation of an input sentence). The key contributions are:

1. We incorporated a discriminative model in a phrase-based decoder. We obtained comparable results with the state-of-art phrase-based decoder (see section 6.1). The advantage in using our approach is that it has the flexibility to incorporate richer contextual and linguistic feature functions.

2. We show that the incorporation of syntactic information (POS tags) in our discriminative model boosted the performance of translation. The lexical accuracy using our approach improved by **6.1%** when syntactic features were used in addition to the lexical features. Similarly, the BLEU score improved by **2.3** points when syntactic features were used compared to the model that uses lexical features alone. The accuracies are likely to improve when richer linguistic feature functions (that use parse structure) are incorporated in our approach.

In future, we plan to work on:

1. Experiment with rich syntactic and structural features (parse tree-based features) using our approach.

2. Experiment on other language pairs such as Arabic-English and Hindi-English.

3. Improving LCS decoding algorithm using syntactic cues in the target (Venkatapathy and Bangalore, 2007) such as supertags.

References

Bangalore, S., P. Haffner, and S. Kanthak. 2007. Statistical machine translation through global lexical selection and sentence reconstruction. In *Annual Meeting-Association for Computational Linguistics*, volume 45, page 152.

Berger, A.L., V.J.D. Pietra, and S.A.D. Pietra. 1996. A maximum entropy approach to natural language processing. *Computational linguistics*, 22(1):39–71.

Brown, P.F., V.J.D. Pietra, S.A.D. Pietra, and R.L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics*, 19(2):263–311.

Chen, S.F. and J. Goodman. 1999. An empirical study of smoothing techniques for language modeling. *Computer Speech and Language*, 13(4):359–394.

Haffner, P. 2006. Scaling large margin classifiers for spoken language understanding. *Speech Communication*, 48(3-4):239–261.

Hassan, H., K. Sima'an, and A. Way. 2009. A syntactified direct translation model with linear-time decoding. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3*, pages 1182–1191. Association for Computational Linguistics.

Ittycheriah, A. and S. Roukos. 2007. Direct translation model 2. In *Proceedings of NAACL HLT*, pages 57–64.

Koehn, P. and H. Hoang. 2007. Factored translation models. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 868–876.

Koehn, P., F.J. Och, and D. Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 48–54. Association for Computational Linguistics.

Koehn, P., H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Annual meeting-association for computational linguistics*, volume 45, page 2.

Och, F.J. and H. Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of ACL*, volume 2, pages 295–302.

Och, F.J., C. Tillmann, H. Ney, et al. 1999. Improved alignment models for statistical machine translation. In *Proc. of the Joint SIGDAT Conf. on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 20–28.

Papineni, KA, S. Roukos, and RT Ward. 1998. Maximum likelihood and discriminative training of directtranslation models. In *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*, volume 1.

Papineni, K., S. Roukos, T. Ward, and W.J. Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Venkatapathy, S. and S. Bangalore. 2007. Three models for discriminative machine translation using Global Lexical Selection and Sentence Reconstruction. In *Proceedings of the NAACL-HLT 2007/AMTA Workshop on Syntax and Structure in Statistical Translation*, pages 96–102. Association for Computational Linguistics.

Venkatapathy, Sriram and Srinivas Bangalore. 2009. Discriminative Machine Translation Using Global Lexical Selection. *ACM Transactions on Asian Language Information Processing*, 8(2).

Xiong, D., Q. Liu, and S. Lin. 2006. Maximum entropy based phrase reordering model for statistical machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, page 528. Association for Computational Linguistics.